

Algorithms based on DQM with new sets of base functions for solving parabolic partial differential equations in $(2 + 1)$ dimension

Brajesh Kumar Singh*, Pramod Kumar

Department of Applied Mathematics, School of Physical Sciences,
Babasaheb Bhimrao Ambedkar University, Lucknow 226025 Uttar Pradesh INDIA
bksingh0584@gmail.com

Abstract. This paper deals with the numerical computations of two space dimensional time dependent *parabolic partial differential equations* by adopting an optimal five stage fourth-order strong stability preserving Runge - Kutta (SSP-RK54) scheme for time discretization, and three methods of differential quadrature with different sets of modified B-splines as base functions, for space discretization: namely- *i*) mECDQM: (DQM with modified extended cubic B-splines); *ii*) mExp-DQM: DQM with modified exponential cubic B-splines, and *iii*) MTB-DQM: DQM with modified trigonometric cubic B-splines. Specially, we implement these methods on *convection-diffusion* equation to convert them into a system of first order ordinary differential equations (ODEs), in time. The resulting system of ODEs can be solved using any time integration method, while we prefer SSP-RK54 scheme. All the three methods are found stable for two space convection-diffusion equation by employing matrix stability analysis method. The accuracy and validity of the methods are confirmed by three test problems of two dimensional *convection-diffusion* equation, which shows that the proposed approximate solutions by any of the method are in good agreement with the exact solutions.

Keywords: Convection-diffusion equation, modified trigonometric cubic-B-splines, modified exponential cubic-B-splines, modified extended cubic-B-splines, differential quadrature method, SSP-RK54 scheme, Thomas algorithm

1 Introduction

The convection-diffusion model can be expressed mathematically, which is a semi linear parabolic partial differential equation. Specially, we consider an initial value system of *convection-diffusion* equation in 2 dimension as:

$$\left. \begin{aligned} \frac{\partial u}{\partial t} - \alpha_x \frac{\partial^2 u}{\partial x^2} - \alpha_y \frac{\partial^2 u}{\partial y^2} + \beta_x \frac{\partial u}{\partial x} + \beta_y \frac{\partial u}{\partial y} &= 0, \\ u(x, y, 0) &= u^0(x, y), \end{aligned} \right\} \text{in } \Omega \times (0, T], \quad (1)$$

together with the Dirichlet boundary conditions:

$$\left. \begin{aligned} u(a, y, t) &= f_1(y, t), & u(b, y, t) &= f_2(y, t), \\ u(x, c, t) &= f_3(x, t), & u(x, d, t) &= f_4(x, t), \end{aligned} \right\} \text{on } \partial\Omega \times (0, T], \quad (2)$$

or Neumann boundary conditions:

$$\left. \begin{aligned} \frac{\partial u}{\partial x} \Big|_{x=a,y} &= g_1(y, t), & \frac{\partial u}{\partial x} \Big|_{x=b,y} &= g_2(y, t), \\ \frac{\partial u}{\partial y} \Big|_{x,y=c} &= g_3(x, t), & \frac{\partial u}{\partial y} \Big|_{x,y=d} &= g_4(x, t), \end{aligned} \right\} \text{on } \partial\Omega \times (0, T]. \quad (3)$$

* Address for Correspondence: Department of Applied Mathematics, School of Physical Sciences, Babasaheb Bhimrao Ambedkar University, Lucknow 226025 Uttar Pradesh INDIA

where $\partial\Omega$ is the boundary of computational domain $\Omega = [a, b] \times [c, d] \subset \mathbb{R}^2$, $(0, T]$ is time interval, and f_i, g_i ($i = 1, 2, 3, 4$) and u_0 are known smooth functions, and $u(x, y, t)$ denote heat or vorticity. The parameters: β_x and β_y are constant convective velocities while the constants $\alpha_x > 0, \alpha_y > 0$ are diffusion coefficients in the direction of x and y , respectively.

The convection-diffusion models have remarkable applications in various branches of science and engineering, for instance, fluid motion, heat transfer, astrophysics, oceanography, meteorology, semiconductors, hydraulics, pollutant and sediment transport, and chemical engineering. Specially, in computational hydraulics and fluid dynamics to model convection-diffusion of quantities such as mass, heat, energy, vorticity [26]. Many researchers have paid their attention to develop some schemes which could produce accurate, stable and efficient solutions behavior of convection-diffusion problems, see [36,38,39,40,41,42] and the references therein.

In the last years, the convection-diffusion equation (1) has been solved numerically using various techniques: namely- finite element method [13], Lattice Boltzmann method [9], finite-difference scheme and higher-order compact finite difference schemes [15,21,23,31,34]. A nine-point high-order compact implicit scheme proposed by Noye and Tan [22] is third-order accurate in space and second-order accurate in time, and has a large zone of stability. An extension of higher order compact difference techniques for steady-state [23] to the time-dependent problems have been presented by Spitz and Carey [29], are fourth-order accurate in space and second or lower order accurate in time but conditionally stable. The fourth-order compact finite difference unconditionally stable scheme due to Dehghan and Mohebbi [12] have the accuracy of order $O(h^4, t^4)$. A family of unconditionally stable finite difference schemes presented in [7] have the accuracy of order $O(h^3, t^2)$. The schemes presented in [17] are based on high-order compact scheme and weighted time discretization, are second or lower order accurate in time and fourth-order accurate in space. The high-order alternating direction implicit (ADI) scheme with accuracy of order $O(h^4, t^2)$ proposed by Karaa and Zhang [18], is unconditionally stable. A high-order unconditionally stable exponential scheme for unsteady 1D convection-diffusion equation by Tian and Yua [32] have the accuracy of order $O(h^4, t^2)$. A rational high-order compact alternating direction implicit (ADI) method have been developed for solving 2D unsteady convection-diffusion problems [30] is unconditionally stable and have the accuracy of order $O(h^4, t^2)$. A unconditionally stable fourth-order compact finite difference approximation for discretizing spatial derivatives and the cubic C_1 - spline collocation method in time, proposed by Mohebbi and Dehghan [20], have the accuracy of order $O(h^4, t^4)$. An unconditionally stable, semi-discrete based on Pade approximation, by Ding and Zhang [14], is fourth-order accurate in space and in time both. The most of schemes are based on the two-level finite difference approximations with Dirichlet conditions, and very few schemes have been developed to solve the convection-diffusion equation with Neumann's boundary conditions, see [6,33] and references therein. The fourth-order compact finite difference scheme by Cao et al. [6] is of 5th-order accurate in time and 4th-order in the space. A high-order alternating direction implicit scheme based on fourth-order Pade approximation developed by You [33] is unconditionally stable with the accuracy of order $O(h^4, t^2)$.

The differential quadrature method (DQM) dates back to Bellman et al. [5]. After the seminal paper of Bellman, various test functions have been proposed, among others, spline functions, sinc function, Lagrange interpolation polynomials, radial base functions, modified cubic B-splines, see [19,24,25,27,28,1,2,3,4], etc. Shu and Richards [27] have generalized approach of DQM for numerical simulation of incompressible Navier-Stokes equation. The main goal of this paper is to find numerical solution of initial value system of 2D *convection-diffusion* equation with both kinds of boundary conditions (Dirichlet boundary conditions and Neumann boundary conditions), approximated by DQM with new sets of modified cubic B-splines (modified extended cubic B-splines,

modified exponential cubic B-splines, modified trigonometric cubic B-splines) as base functions, and so called modified trigonometric cubic-B-spline differential quadrature method (MTB-DQM), modified exponential cubic-B-spline differential quadrature method (mExp-DQM) and third modified extended cubic-B-spline differential quadrature method (mECDQ). These methods are used to transform the convection diffusion problem into a system of first order ODEs, in time. The resulting system of ODEs can be solved by using various time integration algorithm, among them, we prefer SSP-RK54 scheme [45,46] due to its reduce storage space, which results in less accumulation errors. The accuracy and adaptability of the method is illustrated by three test problems of two dimensional convection diffusion equations.

The rest of the paper is organized into five more sections, which follow this introduction. Specifically, Section 2 deals with the description of the methods: namely- MTB-DQM, mExp-DQM and mECDQ. Section 3 is devoted to the procedure for the implementation of describe above these methods for the system (1) together with the boundary conditions as in (2) and (3). Section 4 deals with the stability analysis of the methods. Section 5 deals with the main goal of the paper is the numerical computation of three test problems. Finally, Section 6 concludes the results.

2 Description of the methods

The differential quadrature method is an approximation to derivatives of a function is the weighted sum of the functional values at certain nodes [5]. The weighting coefficients of the derivatives is depend only on grids [27]. This is the reason for taking the partitions $P[\Omega]$ of the problem domain $\Omega = \{(x, y) \in R^2 : 0 \leq x, y \leq 1\}$ distributed uniformly as follows:

$$P[\Omega] = \{(x_i, y_j) \in \Omega : h_x = x_{i+1} - x_i, h_y = y_{j+1} - y_j, i \in \Delta_x, j \in \Delta_y\},$$

where $\Delta_x = \{1, 2, \dots, N_x\}$, $\Delta_y = \{1, 2, \dots, N_y\}$, and $h_x = \frac{1}{N_x-1}$ and $h_y = \frac{1}{N_y-1}$ are the discretization steps in both x and y directions, respectively. That is, a uniform partition in each x, y -direction with the following grid points:

$$0 = x_1 < x_2 < \dots < x_i < \dots < x_{N_x-1} < x_{N_x} = 1,$$

$$0 = y_1 < y_2 < \dots < y_j < \dots < y_{N_y-1} < y_{N_y} = 1.$$

Let (x_i, y_j) be the generic grid point and

$$u_{ij} \equiv u_{ij}(t) \equiv u(x_i, y_j, t), \quad i \in \Delta_x, j \in \Delta_y.$$

The r -th order derivative of $u(x, y, t)$, for $r \in \{1, 2\}$, with respect to x, y at (x_i, y_j) for $i \in \Delta_x, j \in \Delta_y$ is approximated as follows:

$$\begin{aligned} \left(\frac{\partial^r u}{\partial x^r} \right)_{ij} &= \sum_{\ell=1}^{N_x} a_{i\ell}^{(r)} u_{\ell j}, & i \in \Delta_x, \quad i = 1, 2, \dots, N \\ \left(\frac{\partial^r u}{\partial y^r} \right)_{ij} &= \sum_{\ell=1}^{N_y} b_{j\ell}^{(r)} u_{i\ell}, & j \in \Delta_y, \quad i = 1, 2, \dots, M \end{aligned} \tag{4}$$

where the coefficients $a_{ip}^{(r)}$ and $b_{jp}^{(r)}$, the time dependent unknown quantities, are termed as the weighting functions of the r th-order derivative, to be computed using various type of base functions.

2.1 The MTB-DQM

The trigonometric cubic B-spline function $T_i = T_i(x)$ at node i in x direction, read as [35,43]:

$$T_i = \frac{1}{\omega} \begin{cases} p^3(x_i), & x \in [x_i, x_{i+1}) \\ p(x_i)\{p(x_i)q(x_{i+2}) + p(x_{i+1})q(x_{i+3})\} + p^2(x_{i+1})q(x_{i+4}), & x \in [x_{i+1}, x_{i+2}) \\ q(x_{i+4})\{p(x_{i+1})q(x_{i+3}) + p(x_{i+2})q(x_{i+4})\} + p(x_i)q^2(x_{i+3}), & x \in [x_{i+2}, x_{i+3}) \\ q^3(x_{i+4}), & x \in [x_{i+3}, x_{i+4}) \end{cases} \quad (5)$$

where $p(x_i) = \sin(\frac{x-x_i}{2})$; $q(x_i) = \sin(\frac{x_i-x}{2})$, $\omega = \sin(\frac{h_x}{2}) \sin(h_x) \sin(\frac{3h_x}{2})$. Then the set $\{T_0, \dots, T_{N_x}, T_{N_x+1}\}$ forms a base over the interval $[a, b]$. Setting

$$a_1 = \frac{\sin^2(\frac{h_x}{2})}{\sin(h_x) \sin(\frac{3h_x}{2})}; \quad a_2 = \frac{2}{1 + 2 \cos(h_x)}; \quad a_4 = \frac{3}{4 \sin(\frac{3h_x}{2})} = -a_3;$$

$$a_5 = \frac{3 + 9 \cos(h_x)}{16 \sin^2(\frac{h_x}{2}) (2 \cos(\frac{h_x}{2}) + \cos(\frac{3h_x}{2}))}; \quad a_6 = \frac{3 \cos^2(\frac{h_x}{2})}{\sin^2(\frac{h_x}{2}) (2 + 4 \cos(h_x))}.$$

The values of T_i and its first and second derivatives in the grid point x_j , denoted by $T_{ij} := T_i(x_j)$, $T'_{ij} := T'_i(x_j)$ and $T''_{ij} := T''_i(x_j)$, respectively, read:

$$T_{ij} = \begin{cases} a_2, & \text{if } i - j = 0 \\ a_1, & \text{if } i - j = \pm 1 \\ 0, & \text{otherwise} \end{cases}; \quad T'_{ij} = \begin{cases} a_4, & \text{if } i - j = 1 \\ a_3, & \text{if } i - j = -1 \\ 0, & \text{otherwise} \end{cases}; \quad T''_{ij} = \begin{cases} a_6, & \text{if } i - j = 0 \\ a_5, & \text{if } i - j = \pm 1 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The modified trigonometric cubic B-splines base functions are defined as follows [1,4]:

$$\begin{cases} \tau_1(x) = T_1(x) + 2T_0(x) \\ \tau_2(x) = T_2(x) - T_0(x) \\ \vdots \\ \tau_j(x) = T_j(x), \text{ for } j = 3, 4, \dots, N_x - 2 \\ \vdots \\ \tau_{N_x-1}(x) = T_{N_x-1}(x) - T_{N_x+1}(x) \\ \tau_{N_x}(x) = T_{N_x}(x) + 2T_{N_x+1}(x) \end{cases} \quad (7)$$

Now, the set $\{\tau_1, \tau_2, \dots, \tau_{N_x}\}$ is the base over $[a, b]$. The procedure to define modified trigonometric cubic B-splines in y direction, is followed analogously.

The evaluation of the weighting coefficients $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ ($r = 1, 2$) In order to evaluate the weighting coefficients $a_{ip}^{(1)}$ of first order partial derivative in Eq. (4), we use the modified trigonometric cubic B-spline $\tau_p(x)$, $p \in \Delta_x$ in DQ method as base functions. Setting $\tau'_{pi} := \tau'_p(x_i)$ and $\tau_{p\ell} := \tau_p(x_\ell)$. Using MTB-DQM, the approximate values of first order derivative is obtained as follows:

$$\tau'_{pi} = \sum_{\ell=1}^{N_x} a_{i\ell}^{(1)} \tau_{p\ell}, \quad p, i \in \Delta_x. \quad (8)$$

Setting $\mathfrak{S} = [\tau_{p\ell}]$, $A = [a_{i\ell}^{(1)}]$, and $\mathfrak{S}' = [\tau'_{p\ell}]$. Eq. (8) reduced to compact matrix form:

$$\mathfrak{S}A^T = \mathfrak{S}'. \quad (9)$$

The coefficient matrix \mathfrak{S} of order N_x can be read from (6) and (7) as:

$$\mathfrak{S} = \begin{bmatrix} a_2 + 2a_1 & a_1 & & & & & \\ & 0 & a_2 & a_1 & & & \\ & & a_1 & a_2 & a_1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_1 & a_2 & a_1 \\ & & & & & a_1 & a_2 & 0 \\ & & & & & & a_1 & a_2 + 2a_1 \end{bmatrix}$$

and the columns of the matrix \mathfrak{S}' read as:

$$\mathfrak{S}'[1] = \begin{bmatrix} 2a_4 \\ a_3 - a_4 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \mathfrak{S}'[2] = \begin{bmatrix} a_4 \\ 0 \\ a_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, \mathfrak{S}'[N_x - 1] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_4 \\ 0 \\ a_3 \end{bmatrix}, \text{ and } \mathfrak{S}'[N_x] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 2a_4 \\ a_3 - a_4 \end{bmatrix}.$$

2.2 The mExp-DQM

The exponential cubic B-splines function $\zeta_i = \zeta_i(x)$ at node i in x direction, reads [40,41]:

$$\zeta_i = \frac{1}{h_x^3} \begin{cases} b_2\{(x_{i-2} - x) - \frac{1}{p} \sinh(p(x_{i-2} - x))\}, & x \in [x_{i-2}, x_{i-1}) \\ a_1 + b_1(x_i - x) + c_1 \exp(p(x_i - x)) + d_1 \exp(p(x_i - x)), & x \in [x_{i-1}, x_i) \\ a_1 + b_1(x - x_i) + c_1 \exp(p(x - x_i)) + d_1 \exp(p(x - x_i)), & x \in [x_i, x_{i+1}) \\ b_2\{(x - x_{i+2}) - \frac{1}{p} \sinh(p(x - x_{i+2}))\}, & x \in [x_{i+1}, x_{i+2}) \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

where

$$a_1 = \frac{pch_x}{pch_x - s}; b_1 = \frac{p}{2} \left(\frac{s^2 - c(1 - c)}{(pch_x - s)(1 - c)} \right), b_2 = \frac{p}{2(pch_x - s)}, c = \cosh(ph_x), s = \sinh(ph_x), \\ c_1 = \frac{1}{4} \left\{ \frac{\exp(-ph_x)(1 - c) + s(\exp(-ph_x) - 1)}{(pch_x - s)(1 - c)} \right\}, d_1 = \frac{1}{4} \left\{ \frac{\exp(ph_x)(c - 1) + s(\exp(ph_x) - 1)}{(pch_x - s)(1 - c)} \right\}.$$

The set $\{\zeta_0, \zeta_1, \zeta_2, \dots, \zeta_{N_x}, \zeta_{N_x+1}\}$ forms a base over $[a, b]$. Setting the values of ζ_i and its first and second derivatives at x_j by $\zeta_{ij} := \zeta_i(x_j)$, $\zeta'_{ij} := \zeta'_i(x_j)$ and $\zeta''_{ij} := \zeta''_i(x_j)$, respectively. Then

$$\zeta_{ij} = \begin{cases} 1, & \text{if } i - j = 0 \\ \frac{s - ph}{2(pch_x - s)}, & \text{if } i - j = \pm 1 \\ 0, & \text{otherwise} \end{cases}; \zeta'_{ij} = \begin{cases} -\frac{p(1-c)}{2(pch_x - s)}, & \text{if } i - j = 1 \\ \frac{p(1-c)}{2(pch_x - s)}, & \text{if } i - j = -1 \\ 0, & \text{otherwise} \end{cases}; \zeta''_{ij} = \begin{cases} -\frac{p^2 s}{2(pch_x - s)}, & \text{if } i - j = 0 \\ \frac{p^2 s}{2(pch_x - s)}, & \text{if } i - j = \pm 1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The modified exponential cubic B-splines base functions are read as:

$$\begin{cases} \varsigma_1(x) = \zeta_1(x) + 2\zeta_0(x) \\ \varsigma_2(x) = \zeta_2(x) - \zeta_0(x) \\ \vdots \\ \varsigma_j(x) = \zeta_j(x), \text{ for } j = 3, 4, \dots, N_x - 2 \\ \vdots \\ \varsigma_{N_x-1}(x) = \zeta_{N_x-1}(x) - \zeta_{N_x+1}(x) \\ \varsigma_{N_x}(x) = \zeta_{N_x}(x) + 2\zeta_{N_x+1}(x) \end{cases} \quad (12)$$

The set $\{\varsigma_1, \varsigma_2, \dots, \varsigma_{N_x}\}$ is a base over $[a, b]$. The procedure to define modified trigonometric cubic B-splines in y direction, is followed analogously.

The evaluation of the weighting coefficients $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ ($r = 1, 2$) Setting $\varsigma'_{pi} := \varsigma'_p(x_i)$ and $\varsigma_{p\ell} := \varsigma_p(x_\ell)$ for all $p, \ell \in \Delta_x$. Using mExp-DQM, the approximate values of the first-order derivative is given by

$$\varsigma'_{pi} = \sum_{\ell=1}^{N_x} a_{i\ell}^{(1)} \varsigma_{p\ell}, \quad p, i \in \Delta_x. \quad (13)$$

Setting $\mathfrak{I} = [\varsigma_{p\ell}]$, $A = [a_{i\ell}^{(1)}]$, and $\mathfrak{I}' = [\varsigma'_{p\ell}]$, then Eq. (13) can be reduced to compact matrix form:

$$\mathfrak{I}A^T = \mathfrak{I}'. \quad (14)$$

Let $\omega = \frac{p(1-c)h_x}{pch_x-s}$ and $\theta = \frac{s-ph_x}{2(pch_x-s)}$. Using Eqns. (11) and (12), the coefficient matrix \mathfrak{I} of order N_x , read as:

$$\mathfrak{I} = \begin{bmatrix} \omega & \theta & & & & & \\ 0 & 1 & \theta & & & & \\ & \theta & 1 & \theta & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \theta & 1 & \theta & \\ & & & & \theta & 1 & 0 \\ & & & & & \theta & \omega \end{bmatrix}$$

and the columns of the matrix \mathfrak{I}' read:

$$\mathfrak{I}'[1] = \begin{bmatrix} \omega/h_x \\ -\omega/h_x \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \mathfrak{I}'[2] = \begin{bmatrix} \omega/2h_x \\ 0 \\ -\omega/2h_x \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, \mathfrak{I}'[N_x - 1] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \omega/2h_x \\ 0 \\ -\omega/2h_x \end{bmatrix}, \text{ and } \mathfrak{I}'[N_x] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \omega/h_x \\ -\omega/h_x \end{bmatrix}.$$

2.3 The mECDQ method

The extended cubic B-splines function $\varphi_i = \varphi_i(x)$, in the x direction and at the knots, reads [42,36,37]:

$$\varphi_i = \frac{1}{24} \begin{cases} 4(1-\lambda)P_{i-2}^3(x) + 3\lambda P_{i-2}^4(x), & x \in [x_{i-2}, x_{i-1}) \\ 24p + 12P_{i-1}^3(x) + 6(2+\lambda)P_{i-1}^2(x) - 12P_{i-1}^3(x) - 3\lambda P_{i-1}^4(x), & x \in [x_{i-1}, x_i) \\ 24p - 12P_{i+1}^3(x) + 6(2+\lambda)P_{i+1}^2(x) + 12P_{i+1}^3(x) - 3\lambda P_{i+1}^4(x), & x \in [x_i, x_{i+1}) \\ 4(\lambda-1)P_{i+2}^3(x) + 3\lambda P_{i+2}^4(x), & x \in [x_{i+1}, x_{i+2}) \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

where $h_x P_i(x) = (x-x_i)$ and $24\varphi = 4-\lambda$, and λ is a free parameter [36]. The set $\{\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_{N_x}, \varphi_{N_x+1}\}$ forms a base over $[a, b]$. Let $12h_x\theta = 8+\lambda$ and $2h_x^2\omega = 2+\lambda$, then the values of φ_i and its first and second derivatives in the grid point x_j , denoted by $\varphi_{ij} := \varphi_i(x_j)$, $\varphi'_{ij} := \varphi'_i(x_j)$ and $\varphi''_{ij} := \varphi''_i(x_j)$, respectively, read:

$$\varphi_{ij} = \begin{cases} \theta, & \text{if } i-j=0 \\ \emptyset, & \text{if } i-j=\pm 1 \\ 0, & \text{otherwise} \end{cases} ; \quad 2h_x\varphi'_{ij} = \begin{cases} 1, & \text{if } i-j=1 \\ -1, & \text{if } i-j=-1 \\ 0, & \text{otherwise} \end{cases} ; \quad \varphi''_{ij} = \begin{cases} -2\omega, & \text{if } i-j=0 \\ \omega, & \text{if } i-j=\pm 1 \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

The modified extended cubic B-splines base functions are defined as follows [36]:

$$\begin{cases} \phi_1(x) = \varphi_1(x) + 2\varphi_0(x) \\ \phi_2(x) = \varphi_2(x) - \varphi_0(x) \\ \vdots \\ \phi_j(x) = \varphi_j(x), \text{ for } j = 3, 4, \dots, N_x - 2 \\ \vdots \\ \phi_{N_x-1}(x) = \varphi_{N_x-1}(x) - \varphi_{N_x+1}(x) \\ \phi_{N_x}(x) = \varphi_{N_x}(x) + 2\varphi_{N_x+1}(x) \end{cases} \quad (17)$$

The set $\{\phi_1, \phi_2, \dots, \phi_{N_x}\}$ is a base over $[a, b]$.

The evaluation of the weighting coefficients $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ ($r = 1, 2$) Setting $\phi'_{pi} := \phi'_p(x_i)$ and $\phi_{p\ell} := \phi_p(x_\ell)$. Using mECDQ method, the approximate values of the first-order derivative is given by

$$\phi'_{pi} = \sum_{\ell=1}^{N_x} a_{i\ell}^{(1)} \phi_{p\ell}, \quad p, i \in \Delta_x. \quad (18)$$

Setting $\Phi = [\phi_{p\ell}]$, $A = [a_{i\ell}^{(1)}]$, and $\Phi' = [\phi'_{pi}]$. Eq. (18) can be re-written in compact matrix form as:

$$\Phi A^T = \Phi'. \quad (19)$$

Using Eqns. (16) and (17), the matrix Φ of order N_x read as:

$$\Phi = \begin{bmatrix} 1 & \wp & & & & & \\ 0 & \theta & \wp & & & & \\ & \wp & \theta & \wp & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \wp & \theta & \wp & \\ & & & & \wp & \theta & \wp \\ & & & & & \wp & 1 \end{bmatrix}$$

and the columns of the matrix Φ' read:

$$\Phi'[1] = \begin{bmatrix} -1/h_x \\ 1/h_x \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \Phi'[2] = \begin{bmatrix} -1/2h_x \\ 0 \\ 1/2h_x \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, \Phi'[N_x - 1] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1/2h_x \\ 0 \\ 1/2h_x \end{bmatrix}, \text{ and } \Phi'[N_x] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1/h_x \\ 1/h_x \end{bmatrix}.$$

Using ‘‘Thomas algorithm’’ the system (9), (14) and (19) have been solved for the weighting coefficients $a_{i1}^{(1)}, a_{i2}^{(1)}, \dots, a_{iN_x}^{(1)}$, for all $i \in \Delta_x$.

Similarly, the weighting coefficients $b_{ij}^{(1)}$, in either case, can be computed by employing these modified cubic B-splines in the y direction.

Using $a_{ij}^{(1)}$ and $b_{ij}^{(1)}$, the weighting coefficients, $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ (for $r \geq 2$) can be computed using the Shu’s recursive formulae [28]:

$$\begin{cases} a_{ij}^{(r)} = r \left(a_{ij}^{(1)} a_{ii}^{(r-1)} - \frac{a_{ij}^{(r-1)}}{x_i - x_j} \right), i \neq j : i, j \in \Delta_x, \\ a_{ii}^{(r)} = - \sum_{i=1, i \neq j}^{N_x} a_{ij}^{(r)}, i = j : i, j \in \Delta_x. \\ b_{ij}^{(r)} = r \left(b_{ij}^{(1)} b_{ii}^{(r-1)} - \frac{b_{ij}^{(r-1)}}{y_i - y_j} \right), i \neq j : i, j \in \Delta_y \\ b_{ii}^{(r)} = - \sum_{i=1, i \neq j}^{N_y} b_{ij}^{(r)}, i = j : i, j \in \Delta_y. \end{cases} \quad (20)$$

where r denote r -th order spatial derivative. In particular, the weighting coefficients $a_{ij}^{(2)}, b_{ij}^{(2)}$ of order 2 can be obtained by taking $r = 2$ in (20).

3 Implementation of the method for 2D convection-diffusion equation

After computing the approximate values of first and second order spatial partial derivatives from one of the above three methods, one can re-write Eq (1) as follows:

$$\begin{cases} \frac{du_{ij}}{dt} = \alpha_x \sum_{k=1}^{N_x} a_{ik}^{(2)} u_{kj} + \alpha_y \sum_{k=1}^{N_y} b_{jk}^{(2)} u_{ik} - \beta_x \sum_{k=1}^{N_x} a_{ik}^{(1)} u_{kj} - \beta_y \sum_{k=1}^{N_y} b_{jk}^{(1)} u_{ik}, \\ u_{ij}(t=0) = u_{ij}^0. \end{cases} \quad (21)$$

In case of the Dirichlet conditions, the solutions on boundaries can directly read from (2) as:

$$u_{1j} = f_1(y_j, t); u_{N_x j} = f_2(y_j, t); u_{i1} = f_3(x_i, t); u_{iN_y} = f_4(x_i, t), \forall i \in \Delta_x, j \in \Delta_y, t \in (0, T]. \quad (22)$$

On the other hand, if the boundary conditions are Neumann or mixed type, then the solutions at the boundary are obtained by using any above methods (MTB-DQM, mExp-DQM or mECDQ method) on the boundary, which gives a system of two equations. On solving it we get the desired solution on the boundary as follows:

From Eq. (4) with $r = 1$ and the Neumann boundary conditions (3) at $x = a$ and $x = b$, we get

$$\left. \begin{aligned} \sum_{k=1}^{N_x} a_{1k}^{(1)} u_{kj} &= g_1(y_j, t), \\ \sum_{k=1}^{N_x} a_{N_x k}^{(1)} u_{kj} &= g_2(y_j, t), \end{aligned} \right\} j \in \Delta_y. \quad (23)$$

In terms of matrix system for $u_{1j}, u_{N_x j}$, the above equation can be rewritten as

$$\begin{bmatrix} a_{11}^{(1)} & a_{1N_x}^{(1)} \\ a_{N_x 1}^{(1)} & a_{N_x N_x}^{(1)} \end{bmatrix} \begin{bmatrix} u_{1j} \\ u_{N_x j} \end{bmatrix} = \begin{bmatrix} S_j^a \\ S_j^b \end{bmatrix}, \quad (24)$$

where $S_j^a = g_1(y_j, t) - \sum_{k=2}^{N_x-1} a_{1k}^{(1)} u_{kj}$ and $S_j^b = g_2(y_j, t) - \sum_{k=2}^{N_x-1} a_{N_x k}^{(1)} u_{kj}$. On solving (24), for the boundary values u_{1j} and $u_{N_x j}, j \in \Delta_y$, we get

$$u_{1j} = \frac{S_j^a a_{N_x N_x}^{(1)} - S_j^b a_{1N_x}^{(1)}}{a_{11}^{(1)} a_{N_x N_x}^{(1)} - a_{N_x 1}^{(1)} a_{1N_x}^{(1)}}, \quad u_{N_x j} = \frac{S_j^b a_{11}^{(1)} - S_j^a a_{N_x 1}^{(1)}}{a_{11}^{(1)} a_{N_x N_x}^{(1)} - a_{N_x 1}^{(1)} a_{1N_x}^{(1)}}. \quad (25)$$

Analogously, for the Neumann boundary conditions (3) at $y = c$ and $y = d$, the solutions for the boundary values u_{i1} and $u_{iN_y}, i \in \Delta_x$ can be obtained as:

$$u_{i1} = \frac{S_i^c b_{N_y N_y}^{(1)} - S_i^d b_{1N_y}^{(1)}}{b_{11}^{(1)} b_{N_y N_y}^{(1)} - b_{N_y 1}^{(1)} b_{1N_y}^{(1)}}, \quad u_{iN_y} = \frac{S_i^d b_{11}^{(1)} - S_i^c b_{N_y 1}^{(1)}}{b_{11}^{(1)} b_{N_y N_y}^{(1)} - b_{N_y 1}^{(1)} b_{1N_y}^{(1)}}, \quad (26)$$

where $S_i^c = g_3(x_i, t) - \sum_{k=2}^{N_y-1} b_{1k}^{(1)} u_{ik}$ and $S_i^d = g_4(x_i, t) - \sum_{k=2}^{N_y-1} b_{N_y k}^{(1)} u_{ik}$.

After implementing the boundary values, Eq (21) can be written in compact matrix form as follows:

$$\begin{cases} \frac{dU}{dt} = BU + F, \\ U(0) = U^0, \end{cases} \quad (27)$$

where

- a) $U = [U_{ij}]$ is the solution vector:
 $U = (u_{22}, u_{23}, \dots, u_{2(N_y-1)}, u_{32}, u_{33}, \dots, u_{3(N_y-1)}, \dots, u_{(N_y-1)2}, u_{(N_y-1)3}, \dots, u_{(N_y-1)(N_y-1)})$, and U^0 represents the initial solution vector.
- b) $F = [F_{ij}]$ is the vector of order $(N_y - 2)(N_y - 2)$ containing the boundary values, i.e.,
 $F_{ij} = \alpha_x (a_{i1}^{(2)} u_{1j} + u_{N_x j}) + \alpha_y (b_{j1}^{(2)} u_{i1} + b_{jN_y}^{(2)} u_{iN_y}) - \beta_x (a_{i1}^{(1)} u_{1j} + a_{iN_x}^{(1)} u_{N_x j}) - \beta_y (b_{j1}^{(1)} u_{i1} + b_{jN_y}^{(1)} u_{iN_y})$.
- c) B be a square matrix of order $(N_x - 2)(N_y - 2)$ given as

$$B = \alpha_x A_2 + \alpha_y B_2 - \beta_x A_1 - \beta_y B_1, \quad (28)$$

where A_r and B_r ($r = 1, 2$) are the block diagonal matrices of the weighting coefficients $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$, respectively as given below

$$A_r = \begin{bmatrix} a_{22}^{(r)} I & a_{23}^{(r)} I & \dots & a_{2(N_x-1)}^{(r)} I \\ a_{32}^{(r)} I & a_{33}^{(r)} I & \dots & a_{3(N_x-1)}^{(r)} I \\ \vdots & \vdots & \ddots & \vdots \\ a_{(N_x-1)2}^{(r)} I & a_{(N_x-1)3}^{(r)} I & \dots & a_{(N_x-1)(N_x-1)}^{(r)} I \end{bmatrix}, \text{ and } B_r = \begin{bmatrix} M_r & O & \dots & O \\ O & M_r & \dots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & M_r \end{bmatrix}, \quad (29)$$

where I and O are the matrices of order $N_y - 2$, and the sub-matrix M_r of the block diagonal matrix B_r is given by

$$M_r = \begin{bmatrix} b_{22}^{(r)} & b_{23}^{(r)} & \dots & b_{2(N_y-1)}^{(r)} \\ b_{32}^{(r)} & b_{33}^{(r)} & \dots & b_{3(N_y-1)}^{(r)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(N_y-1)2}^{(r)} & b_{(N_y-1)3}^{(r)} & \dots & b_{(N_y-1)(N_y-1)}^{(r)} \end{bmatrix}. \quad (30)$$

Finally, we adopted SSP-RK54 scheme [46] to solve the initial value system (27) as:

$$\begin{aligned} U^{(1)} &= U^m + 0.391752226571890 \Delta t L(U^m) \\ U^{(2)} &= 0.444370493651235 U^m + 0.555629506348765 U^{(1)} + 0.368410593050371 \Delta t L(U^{(1)}) \\ U^{(3)} &= 0.620101851488403 U^m + 0.379898148511597 U^{(2)} + 0.251891774271694 \Delta t L(U^{(2)}) \\ U^{(4)} &= 0.178079954393132 U^m + 0.821920045606868 U^{(3)} + 0.544974750228521 \Delta t L(U^{(3)}) \\ U^{m+1} &= 0.517231671970585 U^{(2)} + 0.096059710526147 U^{(3)} + 0.063692468666290 \Delta t L(U^{(3)}) \\ &\quad + 0.386708617503269 U^{(4)} + 0.226007483236906 \Delta t L(U^{(4)}), \end{aligned}$$

where $LU = BU + F$.

4 Stability of the methods for 2D convection-diffusion equation

The stability of the method MTB-DQM for 2D convection-diffusion equation (1) depends on the stability of the initial value system of ODEs as defined in (27). Noticed that whenever the system of ODEs (27) is unstable, the proposed method for temporal discretization may not converge to the exact solution. Moreover, being the exact solution can directly obtained by means of the eigenvalues method, the stability of (27) depends on the eigenvalues of the coefficient matrix B [44]. In fact, the stability region is the set $\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \leq 1, z = \lambda_B \Delta t\}$, where $R(\cdot)$ is the stability

function and λ_B is the eigenvalue of the coefficient matrix B . The stability region for SSP-RK54 scheme is depicted in [36, Fig.1], from which one can claim that for the stability of the system (27) it is sufficient that $\lambda_B \Delta t \in \mathcal{S}$ for each eigenvalue λ_B of the coefficient matrix B . Hence, the real part of each eigenvalue is necessarily either zero or negative.

It is seen that the eigenvalues of the matrices A_r and B_r ($r = 1, 2$) have identical nature. Therefore, it is sufficient to compute the eigenvalues, λ_1 and λ_2 , of the matrices A_1 and A_2 for different values of grid sizes ($N_x \times N_y$). The eigenvalues λ_1 and λ_2 for $N_x = N_y = 11, 21, 31, 41$ has been depicted in Figure 1. Analogously, one can compute the eigenvalues λ_1 and λ_2 using mECDQ method [36] or mExp-DQM. It is seen that in either case, λ_1 and λ_2 have same nature as in Figure 1. Further, we can get from Figure 1 that each eigenvalue λ_B of the matrix B as defined in Eq. (28) is real and negative. This confirms that the proposed methods produces stable solutions for two dimensional convection-diffusion equations.

5 Numerical results and discussions

This section deals with the main goal, the numerical study of three test problems of the initial value system of convection-diffusion equations with both kinds of the boundary conditions has been done by adopting the methods MTB-DQM, mExp-DQM and mECDQ method along with the integration SSP-RK54 scheme. The accuracy and the efficiency of the methods have been measured in terms of the discrete error norms: namely- average |error| norm (L^2 -error norm) and the maximum error (L_∞ error norm).

Problem 1. Consider the initial value system of 2D convection-diffusion equation (1) with $u_0(x, y) = \exp \left\{ -\frac{(x-x_0)^2}{\alpha_x} - \frac{(y-y_0)^2}{\alpha_y} \right\}$, while values of f_i for $1 \leq i \leq 4$ can be extracted from the exact solution

$$u(x, y, t) = \frac{1}{1+4t} \exp \left\{ -\frac{(x-x_0-\beta_x t)^2}{\alpha_x(1+4t)} - \frac{(y-y_0-\beta_y t)^2}{\alpha_y(1+4t)} \right\},$$

where initial condition is a Gaussian pulse with unit height centered at $(x_0, y_0) = (0.5, 0.5)$.

We have computed the numerical solution of Problem 1 for $\beta_x = \beta_y = 0.8$ and different values of α_x, α_y .

For $\Omega = [1, 2]^2$: L_∞ errors and the rate of convergence (ROC) [36] at $t = 1$ for different values of $h^2 = \Delta t$ ($h_x = h_y = h$) has been reported in Table 1 for $\alpha_x = \alpha_y = 0.05, 0.005$. Table 1 confirms that the proposed solutions obtained by these methods are more accurate, and approaching towards exact solutions. The behavior of the approximate solution at time $t = 1$ taking $h = 0.0125, \Delta t = 0.000625$ is depicted in Figure 2 for $\alpha_x = \alpha_y = 0.05$ and in Figure 3 for $\alpha_x = \alpha_y = 0.005$.

For $\Omega = [0, 2]^2$, time $t = 1.25$, and grid space step size $h = 0.025$: the L^2 and L_∞ error norms in the proposed solutions have been compared with the errors in the solutions by various schemes in [31,22,17,18,31,12] in Table 2 for $\Delta t = 0.00625$ and in Table 3 for $\Delta t = 0.0125$. The initial solution and MTB-DQM solutions in $[0, 2]^2$ with the parameter values $\alpha_x = \alpha_y = 0.01, t = 1.25$ have been depicted in Figure 4, and similar behavior is seen from the other two methods. It is evident from the above reports that the proposed results are more accurate as compared to [31,22,17,18,31,12] and approaching towards the exact solutions.

Problem 2. Consider the initial value system (1) of 2D convection-diffusion equation with $\Omega = [0, 1]^2$ with $u_0(x, y) = a(\exp(-c_x x) + \exp(-c_y y))$, where

$$c_x = \frac{-\beta_x \pm \sqrt{\beta_x^2 + 4b\alpha_x}}{2\alpha_x} > 0, \text{ and } c_y = \frac{-\beta_y \pm \sqrt{\beta_y^2 + 4b\alpha_y}}{2\alpha_y} > 0,$$

and the Neumann boundary condition

$$\begin{cases} \left. \frac{\partial u}{\partial x} \right|_{x=0,y} = -ac_x \exp(bt), & \left. \frac{\partial u}{\partial x} \right|_{x=2,y} = -ac_x \exp(bt - c_x), \\ \left. \frac{\partial u}{\partial y} \right|_{x,y=0} = -ac_y \exp(bt), & \left. \frac{\partial u}{\partial y} \right|_{x,y=2} = -ac_y \exp(bt - c_y), \end{cases} \quad (x, y, t) \in \partial\Omega \times (0, T],$$

Table 1. Rate of convergence of L_∞ error norms in Problem 1 in $[1, 2]^2$ with $\alpha_x = \alpha_y = 0.05, 0.005, \beta_x = \beta_y = 0.8$ taking $\Delta t = h^2$ and various grid sizes $N_x \times N_y$ with $N_x = N_y$

$\alpha_x = \alpha_y = 0.005$						
N_x	MTB-DQM		mExp-DQM (p=0.001)		mECDQ ($\lambda = -0.30$)	
	L_∞	ROC	L_∞	ROC	L_∞	ROC
5	2.91E-03		2.91E-03		1.94E-03	
10	6.41E-04	2.18	6.42E-04	2.181	4.93E-04	1.98
20	6.50E-05	3.30	6.51E-05	3.303	6.05E-05	3.03
40	7.19E-06	3.18	7.18E-06	3.180	6.91E-06	3.13
80	7.22E-07	3.31	7.00E-07	3.358	6.99E-07	3.31
$\alpha_x = \alpha_y = 0.05$						
5	4.93E-04		4.97E-04		3.20E-04	
10	4.86E-05	3.34	4.87E-05	3.35	4.39E-05	2.87
20	4.70E-06	3.37	4.70E-06	3.37	4.35E-06	3.34
40	4.51E-07	3.38	4.46E-07	3.40	4.50E-07	3.27
80	4.58E-08	3.30	3.49E-08	3.67	4.37E-08	3.36

Table 2. Comparison L^2 and L_∞ error norms for Problem 1 with $\alpha_x = \alpha_y = 0.01; \beta_x = \beta_y = 0.8$ for $h_x = h_y = 0.025, \Delta t = 0.00625$

Schemes	L^2	L_∞
P-R-ADI [31]	3.109E-04	7.778E-03
Noye & Tang [22]	1.971E-05	6.509E-04
Kalila et al. [17]	1.597E-05	4.447E-04
Kara & Zhang ADI [18]	9.218E-06	2.500E-04
Tang & Ge ADI[31]	9.663E-06	2.664E-04
Dehghan & Mohebbi [12]	9.431E-06	2.477E-04
MTB-DQM	8.026E-12	4.327E-08
mExp-DQM(p=0.0001)	7.030E-12	4.154E-08
mECDQ ($\lambda = -0.004$)	4.504E-12	3.343E-08
MCB-DQM ($\lambda = 0$)	8.269E-12	4.388E-08

Table 3. Comparison L^2 and L_∞ error norms for Problem 1 with $\alpha_x = \alpha_y = 0.01; \beta_x = \beta_y = 0.8$ for $h_x = h_y = 0.025, \Delta t = 0.0125$

Schemes	L^2	L_∞
Noye & Tang [22]	1.430E-05	4.840E-04
Kalila et al.[17]	1.590E-05	4.480E-04
Dehghan & Mohebbi [12]	9.480E-06	2.469E-04
MTB-DQM	8.900E-12	4.501E-08
mExp-DQM(p=0.0001)	9.147E-12	4.315E-08
mECDQ ($\lambda = -0.005$)	4.646E-12	3.852E-08
MCB-DQM ($\lambda = 0$)	9.147E-12	4.561E-08

or the Dirichlet's conditions can be extracted from the exact solution [8]:

$$u(x, y, t) = a \exp(bt) (\exp(-c_x x) + \exp(-c_y y)).$$

The computation by the proposed methods has been done for different values of $\alpha_x = \alpha_y$ and $\beta_x = \beta_y$.

For $\beta_x = \beta_y = 1$, we take $\Delta t = 0.001$ for the solutions at $t = 1$. The rate of convergence and L^2 error norms in the proposed solutions has been compared with that of due to fourth-order compact finite difference scheme [12] for $\alpha_x = \alpha_y = 0.1, 0.01$, in Table 4. It is found that the proposed solutions from either method are more accurate in comparison to [12], and are in good agreement with the exact solutions.

For $\beta_x = \beta_y = -1$, we take $\Delta t = 0.0005$ for the solution at $t = 1$. In Table 5, the L^2 and L_∞ error norms are compared with that obtained by fourth-order compact finite difference scheme [12] for $\beta_x = \beta_y = -1$, and $\alpha_x = \alpha_y = 0.01$. Rate of convergence have been reported in Table 6. From Tables 5 and 6, we found that the proposed solutions are more accurate as compared to the results in [12], while the rate of convergence is linear. The behavior of solutions is depicted in Figure 5 and 6 with $h = 0.025$ for $\alpha_x = \alpha_y = 0.1$ and $\alpha_x = \alpha_y = 0.01$, respectively.

For the same parameter, mentioned above, the numerical solution is obtained by using Neumann conditions and reported in Table 7. The obtained results are in good agreement with the exact solutions, the rate of convergence for L_∞ is quadratic for each method.

Problem 3. The initial value system of 2D convection-diffusion equation (1) with $\Omega = [0, 1]^2$ and $u_0(x, y) = h(x, y)$, and

$$f_1 = h(0, y) - \bar{c}t; \quad f_1 = h(1, y) - \bar{c}t; \quad f_3 = h(x, 0) - \bar{c}t; \quad f_4 = h(x, 1) - \bar{c}t,$$

Table 4. L^2 -error norms and ROC for Problem 2 with $\beta_x = \beta_y = 1$ and $\alpha_x = \alpha_y = 0.10, 0.01$ at $t = 1$

$\alpha_x = \alpha_y = 0.10$ and $\triangle t = 0.0005$														
h	MTB-DQM				mExp-DQM ($p = 1$)				mECDQ ($\lambda = -0.0001$)				CFDS4[12]	
	L_∞	ROC	L^2	ROC	L_∞	ROC	L^2	ROC	L_∞	ROC	L^2	ROC	L^2	ROC
0.2	4.231E-06		4.969E-10		1.714E-05		3.584E-09		4.231E-06		4.999E-10			
0.1	5.155E-07	3.0	2.472E-11	4.3	5.168E-07	5.1	2.473E-11	7.2	5.153E-07	3.0	2.478E-11	4.3	2.289E-10	
0.05	5.108E-08	3.3	9.041E-13	4.8	5.105E-08	3.3	9.044E-13	4.8	5.107E-08	3.3	9.049E-13	4.8	1.621E-11	3.82
0.025	1.147E-08	2.2	1.922E-14	5.6	1.147E-08	2.2	1.922E-14	5.6	1.147E-08	2.4	1.923E-14	5.6	8.652E-13	4.23
$\alpha_x = \alpha_y = 0.01, \triangle t = 0.001$														
0.2	7.4031E-06		1.4907E-09		3.5634E-05		1.5314E-08		2.0585E-06		6.7290E-11			
0.1	1.1460E-06	2.7	1.3103E-10	3.5	1.1421E-06	5.0	1.3148E-10	6.9	1.5919E-07	3.7	1.1624E-12	5.9	2.749E-09	
0.05	1.4578E-07	3.0	7.5377E-12	4.1	1.4582E-07	3.0	7.5396E-12	4.1	1.9012E-08	3.1	5.4939E-14	4.4	2.394E-10	3.52
0.025	1.3902E-08	3.4	2.7129E-13	4.8	1.3904E-08	3.4	2.7132E-13	4.8	6.2220E-09	1.6	9.0456E-15	2.6	1.658E-11	3.85

Table 5. Comparison of the error norms in Problem 2 with $\beta_x = \beta_y = -1$, $\alpha_x = \alpha_y = 0.01$ taking $\Delta t = 0.0005$ with CFDS4[12]

h	MTB-DQM		mExp-DQM, $p = 0.0001$		mECDQ, $\lambda = -1.75$		CFDS4 [12]	
	L_∞	L^2	L_∞	L^2	L_∞	L^2	L_∞	L^2
0.04	4.1718E-03	1.7833E-03	1.7475E-03	7.9168E-04	1.2939E-03	3.6957E-04	1.1826E-01	4.9331E-03
0.02	5.2095E-04	6.8410E-05	5.2097E-04	6.8445E-05	6.3735E-04	6.8622E-05	1.5310E-02	4.1351E-04
0.01	3.2496E-04	2.1508E-05	3.2484E-04	2.1493E-05	2.5394E-04	1.8260E-05	9.4696E-04	2.8405E-05

where

$$h(x, y) = 5 \exp \left\{ -\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4} \right\} + 7 \exp \left\{ -\frac{(9x+1)^2}{50} - \frac{(9y+1)^2}{10} \right\} \\ + 4 \exp \left\{ -\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4} \right\} - 2 \exp \left\{ -(9x-4)^2 - (9y-7)^2 \right\}$$

The distribution of the initial solution is depicted in Figure ?? . The solutions behavior is obtained for the parameter values: $\alpha_x = 0.2, \alpha_y = 0.3, \beta_x = -0.1, \beta_y = 0.2$ and $\Delta t = 0.0005, h_x = h_y = 0.025$, and is depicted in Figure 7 due to MTB-DQM, also we noticed the similar characteristics obtained using mExp-DQM and mECDQ method. The obtained characteristics agreed well as obtained in [12,34].

Table 6. L^2 and L^∞ error norms in Problem 2 with $\beta_x = \beta_y = -1$ and different values of $\alpha_x = \alpha_y$

h	$\alpha_x = \alpha_y = 0.01, \Delta t = 0.0005$				mExp-DQM, $p = 0.0001$				mECDQ, $\lambda = -0.9$			
	L^∞	ROC	L^2	ROC	L^∞	ROC	L^2	ROC	L^∞	ROC	L^2	ROC
0.1	2.6263E-02		3.6277E-02		2.6280E-02		3.6670E-02		2.4464E-02		4.0175E-02	
0.05	6.8144E-03	1.9	4.1447E-03	3.1	6.8144E-03	1.9	4.1339E-03	3.1	3.3580E-03	2.9	1.6264E-03	4.6
0.025	1.1634E-03	2.6	2.7780E-04	3.9	1.0719E-03	2.7	1.9697E-04	4.4	8.1776E-04	2.0	1.0090E-04	4.0
h	$\alpha_x = \alpha_y = 0.1, \Delta t = 0.0005$				$p = 0.0001$				$\lambda = 0.3$			
	L^∞	ROC	L^2	ROC	L^∞	ROC	L^2	ROC	L^∞	ROC	L^2	ROC
0.01	5.0998E-04		3.5191E-03		5.0738E-04		1.0116E-05		7.4180E-04		2.5558E-05	
0.05	1.8776E-04	1.4	1.8193E-03	1.0	1.8776E-04	1.4	2.9979E-06	1.8	2.0972E-04	1.8	5.1381E-06	2.3
0.025	9.9342E-05	0.9	9.1372E-04	1.0	9.9342E-05	0.9	7.9462E-07	1.9	9.5677E-05	1.1	8.1188E-07	2.7

Table 7. The L^2 -error norm in Problem 2 with Neumann boundary conditions for the different values of α_x, α_y and β_x, β_y with $\Delta t = 0.0005$

h	$\alpha_x = \alpha_y = 0.1$ and $\beta_x = \beta_y = -1$											
	MTB-DQM				mExp-DQM, $p = 0.001$				mECDQ, $\lambda = -0.9$			
	L^∞	ROC	L^2	ROC	L^∞	ROC	L^2	ROC	L^∞	ROC	L^2	ROC
0.05	1.0176E-02		6.7504E-03		1.0177E-02		6.7513E-03		9.1549E-03		5.7412E-03	
0.025	2.2164E-03	2.2	1.0428E-03	2.7	2.2165E-03	2.2	1.0429E-03	2.7	1.9834E-03	2.2	8.7214E-04	2.7
0.0125	2.9460E-04	2.9	7.2888E-05	3.8	2.9460E-04	2.9	7.2889E-05	3.8	2.5565E-04	3.0	5.6518E-05	3.9
h	$\alpha_x = \alpha_y = 0.01$ and $\beta_x = \beta_y = -1$											
	$p = 0.001$				$\lambda = -0.9$							
0.05	1.2070E-01		6.3285E-01		1.2070E-01		6.3293E-01		1.0921E-01		6.0454E-01	
0.025	2.7808E-02	2.1	6.2302E-02	3.3	2.7809E-02	2.1	6.2304E-02	3.3	2.6004E-02	2.1	5.5004E-02	3.5
0.0125	6.0348E-03	2.2	6.4433E-03	3.3	6.0348E-03	2.2	6.4433E-03	3.3	5.4733E-03	2.2	5.5668E-03	3.3
h	$\alpha_x = \alpha_y = 0.1$ and $\beta_x = \beta_y = 1$											
	$p = 0.01$				$\lambda = -0.9$							
0.05	1.4931E-05		5.8044E-08		1.4932E-05		5.8056E-08		1.3598E-05		4.8332E-08	
0.025	3.3009E-06	2.2	1.0664E-08	2.4	3.3009E-06	2.2	1.0665E-08	2.4	2.9824E-06	2.2	8.7286E-09	2.5
0.0125	4.5947E-07	2.8	8.0780E-10	3.7	4.5947E-07	2.8	8.0780E-10	3.7	3.9840E-07	2.9	6.0761E-10	3.8
h	$\alpha_x = \alpha_y = 0.01$ and $\beta_x = \beta_y = 1$											
	$p = 0.01$				$\lambda = -0.9$							
0.05	1.7132E-05		6.9905E-08		1.7132E-05		6.9910E-08		1.6511E-05		6.5219E-08	
0.025	3.9931E-06	2.1	1.4135E-08	2.3	3.9932E-06	2.1	1.4136E-08	2.3	3.7892E-06	2.1	1.2756E-08	2.4
0.0125	8.8931E-07	2.2	2.7052E-09	2.4	8.8931E-07	2.2	2.7052E-09	2.4	8.2511E-07	2.2	2.3316E-09	2.5

6 Conclusions

In this paper, the numerical computations of initial value system of two dimensional *convection-diffusion* equations with both kinds of boundary conditions has been done by adopting three methods: modified exponential cubic B-splines DQM, modified trigonometric cubic B-splines DQM, and mECDQ method [36], which transforms the *convection-diffusion* equation into a system of first order ordinary differential equations (ODEs), in time, which is solved by using SSP-RK54 scheme.

The methods are found stable for two space convection-diffusion equation by employing matrix stability analysis method. Section 5 shows that the proposed solutions are more accurate in comparison to the solutions by various existing schemes, and are in good agreement with the exact solutions.

The order of accuracy of the proposed methods for the convection-diffusion problem with Dirichlet's boundary conditions is cubic whenever $\beta_x, \beta_y > 0$ and otherwise it is super linear, in space. On the other hand, the order of accuracy of the proposed methods for the convection-diffusion problem with Neumann boundary condition is quadratic with respect to L_2 error norms, see Table 7.

References

1. G. Arora and B. K. Singh, Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method. *Applied Math Comput* 224 (2013) 166-177.
2. B. K. Singh and G. Arora, A numerical scheme to solve Fisher-type reaction-diffusion equations, *Nonlinear Studies/MESA- MATHEMATICS IN ENGINEERING, SCIENCE AND AEROSPACE* 5(2) (2014) 153-164.
3. B. K. Singh, G. Arora, M. K. Singh, A numerical scheme for the generalized Burgers-Huxley equation, *Journal of the Egyptian Mathematical Society* (2016) <http://dx.doi.org/10.1016/j.joems.2015.11.003>.
4. B. K. Singh and Carlo Bianca, A new numerical approach for the solutions of partial differential equations in three-dimensional space, *Appl. Math. Inf. Sci.* 10, No. 5, 1-10 (2016).
5. R. Bellman, B.G. Kashef and J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear differential equations. *J Comput Phy* 10 (1972) 40-52.
6. Huai-Huo Cao, Li-Bin Liu, Yong Zhang and Sheng-mao Fu, A fourth-order method of the convection-diffusion equations with Neumann boundary conditions, *Applied Mathematics and Computation* 217 (2011) 9133-9141.
7. M.M. Cecchi and M.A. Pirozzi, High order finite difference numerical methods for time-dependent convection-dominated problems, *Appl. Numer. Math.* 55 (2005) 334-356.
8. P.P. Chinchapatnam, K. Djidjeli and P.B. Nair, Unsymmetric and symmetric meshless schemes for the unsteady convection-diffusion equation, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 2432-2453.
9. S. P. Dawson, S. Chen and G. D. Doolen, Lattice Boltzmann computations for reaction-diffusion equations, *J. Chem. Phys.* 98 (2) (1993) 1514-1523.
10. M. Dehghan, Weighted finite difference techniques for the one-dimensional advection-diffusion equation, *Appl. Math. Comput.* 147 (2004) 307-319.
11. M. Dehghan, Numerical solution of the three-dimensional advection-diffusion equation, *Appl. Math. Comput.* 150 (2004) 5-19.
12. M. Dehghan and A. Mohebbi, High-order compact boundary value method for the solution of unsteady convection-diffusion problems, *Math. Comput. Simulat.* 79 (2008) 683-699.
13. S. Dhawan, S. Kapoor and S. Kumar, Numerical method for advection diffusion equation using FEM and B-splines, *Journal of Computational Science* 3 (2012) 429-437.
14. Hengfei Ding and Yuxin Zhang, A new difference scheme with high accuracy and absolute stability for solving convection-diffusion equations, *Journal of Computational and Applied Mathematics* 230 (2009) 600-606.
15. M.M. Gupta, R.P. Manohar and J.W. Stephenson, A single cell high-order scheme for the convection-diffusion equation with variable coefficients, *Int. J. Numer. Methods Fluids* 4 (1984) 641-651.
16. M.K. Jain, *Numerical Solution of Differential Equations*, 2nd Ed., Wiley, New York, NY, 1983.
17. J.C. Kalita, D.C. Dalal and A.K. Dass, A class of higher order compact schemes for the unsteady two-dimensional convection-diffusion equation with variable convection coefficients, *Int. J. Numer. Methods Fluids* 38 (2002) 1111-1131.
18. S. Karaa and J. Zhang, High order ADI method for solving unsteady convection-diffusion problems, *J. Comput. Phys.* 198 (2004) 1-9.

19. Korkmaz A. Shock wave simulations using sinc differential quadrature method. *Int J Comput Aided Engg Software* 28(6) (2011) 654-674.
20. A. Mohebbi and M. Dehghan, High-order compact solution of the one-dimensional heat and advection-diffusion equations, *Applied Mathematical Modelling* 34 (2010) 3071-3084
21. B.J. Noye, A compact unconditionally stable finite-difference method for transient one-dimensional advection-diffusion, *Commun. Appl. Numer. Methods* 7 (1991) 501-512.
22. B.J. Noye and H.H. Tan, Finite difference methods for solving the two-dimensional advection-diffusion equation, *Int. J. Numer. Methods Fluids*, 26 (1988) 1615-1629.
23. W.F. Spitz, High-order Compact Finite Difference Schemes for Computational Mechanics, PhD Thesis, University of Texas at Austin, Austin, TX, 1995.
24. J.R. Quan and C.T. Chang, New insights in solving distributed system equations by the quadrature methods-I. *Comput Chem Eng* 13(1989) 779-788.
25. J.R. Quan and C.T. Chang, New insights in solving distributed system equations by the quadrature methods-II. *Comput Chem Eng* 13 (1989) 1017-1024.
26. P. J. Roach, *Computational Fluid Dynamics*, Hermosa Press, Albuquerque, NM, 1976.
27. C. Shu and B. E. Richards, Application of generalized differential quadrature to solve two dimensional incompressible Navier-Stokes equations. *Int J Numer Meth Fluids* 15 (1992) 791-8.
28. C. Shu, *Differential Quadrature and its Application in Engineering*. Athenaum Press Ltd., Great Britain, 2000.
29. W.F. Spitz and G.F. Carey, Extension of high-order compact schemes to time-dependent problems, *Numer. Methods Partial Diff. Eq.* 17 (2001) 657-672.
30. Zhen F. Tian, A rational high-order compact ADI method for unsteady convection-diffusion equations, *Computer Physics Communications* 182 (2011) 649662.
31. Z.F. Tian and Y.B. Ge, A fourth-order compact ADI method for solving two-dimensional unsteady convection-diffusion problems, *J. Comput. Appl. Math.* 198 (2007) 268-286.
32. Zhen F. Tian and P. X. Yua, A high-order exponential scheme for solving 1D unsteady convection-diffusion equations, *Journal of Computational and Applied Mathematics* 235 (2011) 2477-2491.
33. Donghyun You, A high-order Pade' ADI method for unsteady Convection-diffusion equations, *Journal of Computational Physics* 214 (2006) 1-11
34. M. Zerroukat, K. Djidjeli and A. Charafi, Explicit and implicate messless method for linear advection-diffusion-type partial differential eqations, *International Journal Numerical Method in Engineering* 48 (2000) 19-35.
35. G. Arora, V. Joshi, A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes, *Ain Shams Eng J* (2016), <http://dx.doi.org/10.1016/j.asej.2016.06.013>.
36. B.K. Singh, P. Kumar, A novel approach for numerical computation of Burgers' equation (1 + 1) and (2 + 1) dimension, *Alexandria Eng. J.* (2016), <http://dx.doi.org/10.1016/j.aej.2016.08.023>.
37. B.K. Singh, P. Kumar, An algorithm based on a new DQM with modified extended cubic B-splines for numerical study of two dimensional hyperbolic telegraph equation, *Alexandria Eng. J.* (2016), <http://dx.doi.org/10.1016/j.aej.2016.11.009>.
38. G. Arora, RC Mittal and B. K. Singh, Numerical Solution of BBM-Burger Equation with Quartic B-spline collocation method, *Journal of Engineering Science and Technology, Special Issue* 1, 12/2014, 104 - 116.
39. V. K. Srivastava and B. K. Singh, A Robust finite difference scheme for the numerical solutions of two dimensional time-dependent coupled nonlinear Burgers' equations, *International Journal of Applied Mathematics and Mechanics* 10(7) (2014) 28-39.
40. A. Korkmaz, and H. K. Akmaz, Numerical Simulations for Transport of Conservative Pollutants. *Selcuk Journal of Applied Mathematics* 16(1) (2015).
41. Ozlem Ersoy and Idris Dag, Numerical solutions of the reaction diffusion system by using exponential cubic B-spline collocation algorithms, *Open Phys.* 13(2015)414-427.
42. A. Korkmaz and H.K. Akmaz, Extended B-spline Differential Quadrature Method for Nonlinear Viscous Burgers' Equation, *Proceedings of International Conference on Mathematics and Mathematics Education*, pp 323-323, Elazığ, Turkey 12-14 May, 2016.
43. Muhammad Abbas, Ahmad Abd. Majid, Ahmad Izani Md. Ismail, Abdur Rashid, The application of cubic trigonometric B-spline to the numerical solution of the hyperbolic problems, *Applied Mathematics and Computation* 239 (2014) 74-88.
44. M.K. Jain, *Numerical Solution of Differential Equations*, 2nd ed., Wiley, New York, NY, 1983.
45. S. Gottlieb, D. I. Ketcheson, C. W. Shu, High Order Strong Stability Preserving Time Discretizations, *J. Sci. Comput.* 38 (2009) 251-289.
46. J. R. Spiteri and S. J. Ruuth, A new class of optimal high-order strongstability-preserving time-stepping schemes. *SIAM Journal Numer Anal* 40(2) (2002) 469-491

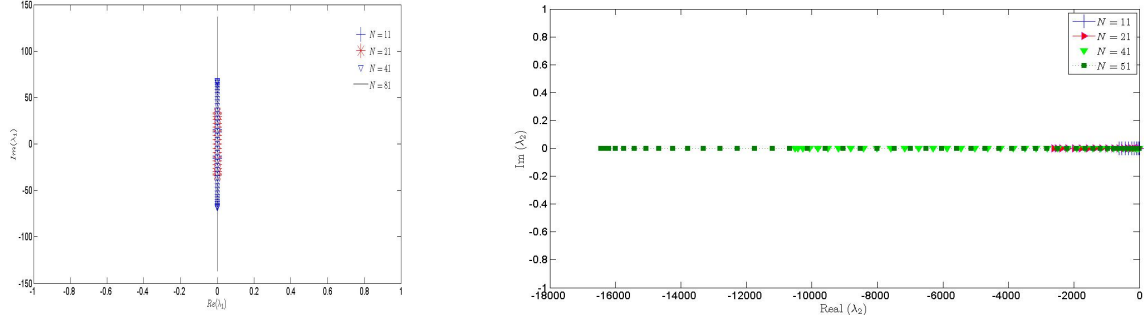


Fig. 1. Eigenvalues λ_1 and λ_2 for different grid sizes

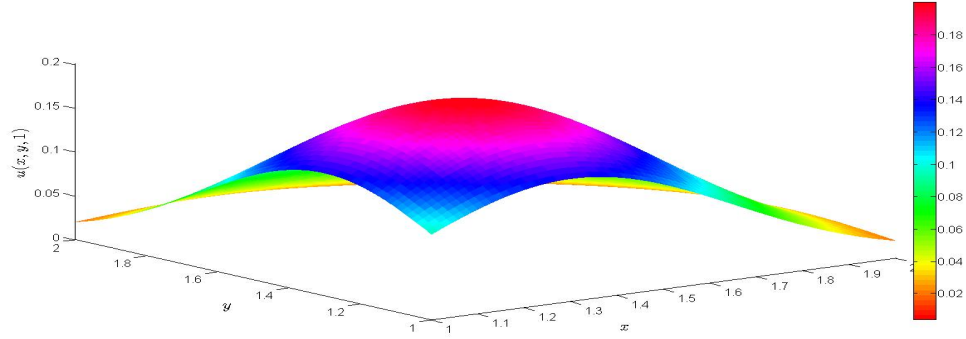


Fig. 2. Surface plot of distribution of 2D convection-diffusion equation (1) with $\alpha_x = \alpha_y = 0.05$, $\beta_x = \beta_y = 0.8$ at $t = 1$

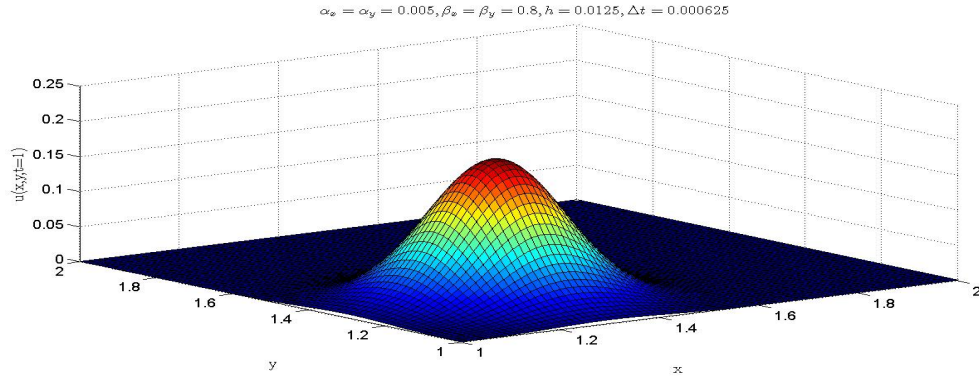


Fig. 3. Surface plot of distribution of 2D convection-diffusion equation (1) with $\alpha_x = \alpha_y = 0.005$, $\beta_x = \beta_y = 0.8$ at $t = 1$

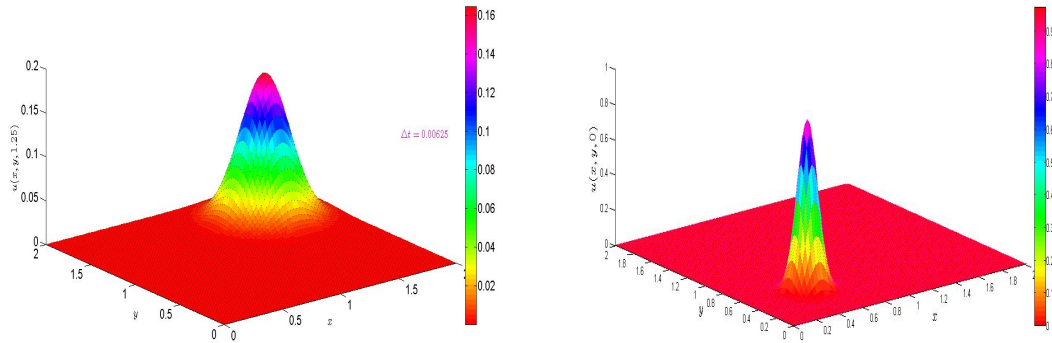


Fig. 4. Surface plot of the initial distribution (right) and the distribution at $t = 1.25$ (left) of 2D convection-diffusion equation (1) with $\alpha_x = \alpha_y = 0.01$, $\beta_x = \beta_y = 0.8$ for $h_x = h_y = 0.025$

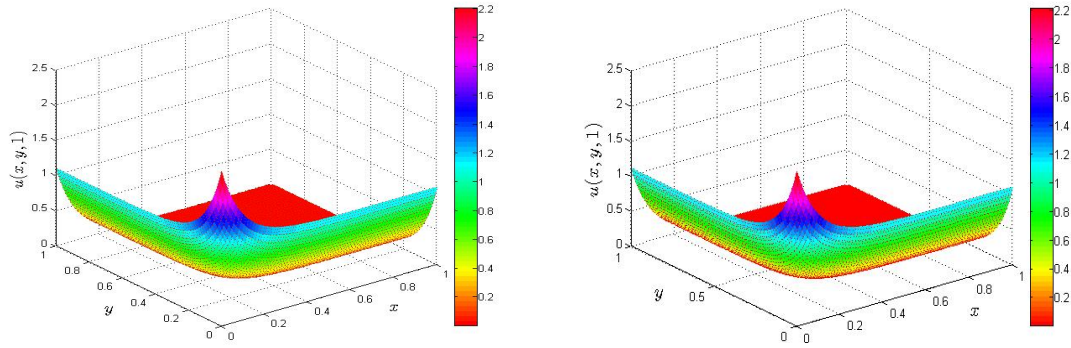


Fig. 5. The approximate (left) and exact solution behavior (right) of 2D convection-diffusion equation (1) with $\alpha_x = \alpha_y = 0.10$, $\beta_x = \beta_y = -1$ at $t = 1$

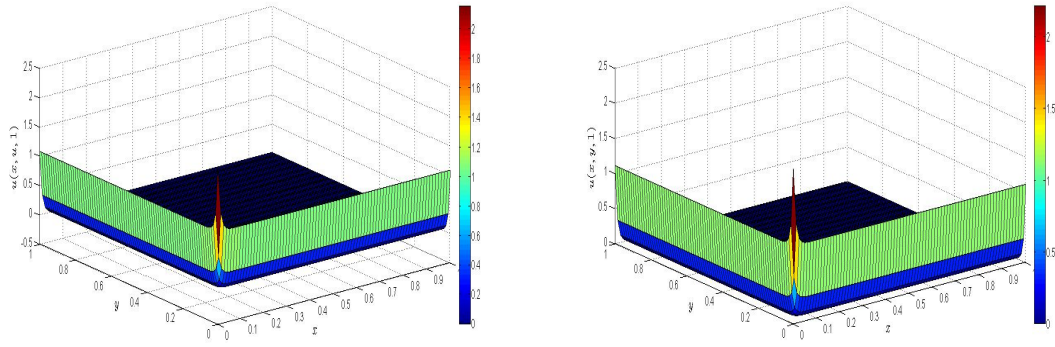


Fig. 6. The approximate solution behavior of 2D convection-diffusion equation (1) with $\alpha_x = \alpha_y = 0.01$, $\beta_x = \beta_y = -1$ at $t = 1$

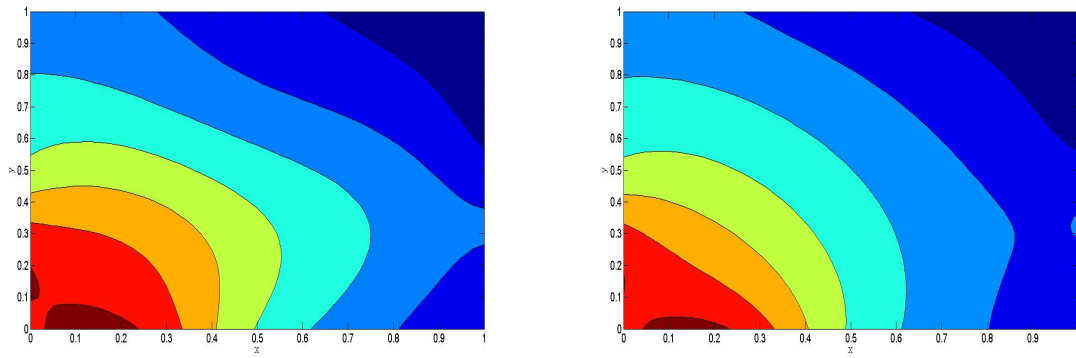


Fig. 7. The approximate solution behavior of the distribution of 2D convection-diffusion equation (1) in Example 3 at $t = 0.1$ (left) and $t = 0.5$ (right)